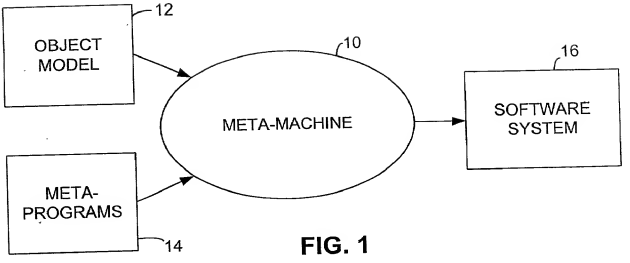
OBJECT MODEL
12
META-MACHINE
10
SOFTWARE SYSTEM
16
META-PROGRAMS
14

FIG. 1

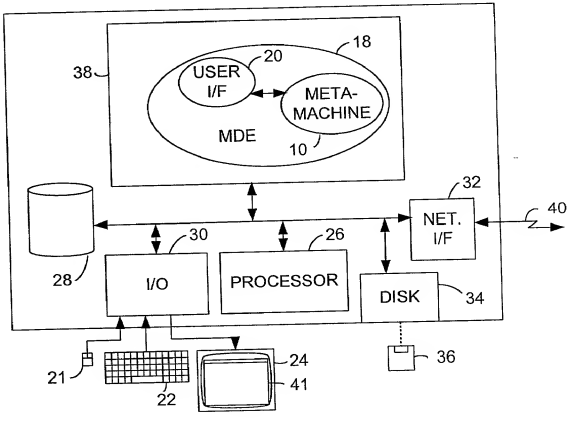
38
18
20
USER I/F
META-MACHINE
MDE
10
32
NET. I/F
40
30
26
28
I/O
PROCESSOR
DISK
34
36
21
22
24
41

FIG. 2

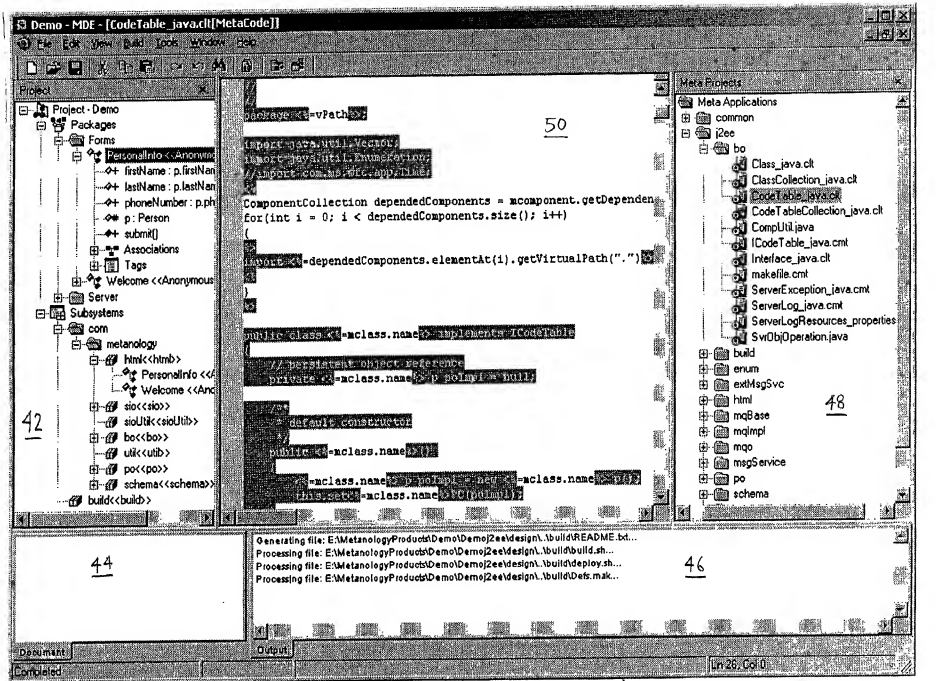
Demo - MDE - [CodeTable_java.clt[MetaCode]]
File Edit View Build Tools Window Help
Project
Project - Demo
Packages
Forms
PersonalInfo <<Anonymous
firstName : p.firstNan
lastName : p.lastNan
phoneNumber : p.phi
p : Person
submit()
Associations
Tags
Welcome <<Anonymous
Server
Subsystems
com
metanology
html<<html>>
PersonalInfo <<A
Welcome <<Anc
sio<<sio>>
sioUtil<<sioUtil>>
bo<<bo>>
util<<util>>
po<<po>>
schema<<schema>>
build<<build>>
42
50
Meta Projects
Meta Applications
common
j2ee
bo
Class_java.clt
ClassCollection_java.clt
CodeTable_java.clt
CodeTableCollection_java.clt
CompUtil.java
ICodeTable_java.cmt
Interface_java.clt
makefile.cmt
ServerException_java.cmt
ServerLog_java.cmt
ServerLogResources_properties
SvrObjOperation.java
build
enum
extMsgSvc
html
mqBase
mqImpl
mqo
msgService
po
schema
48
Generating file: E:\MetanologyProducts\Demo\Demo2ee\design\..\build\README.txt...
Processing file: E:\MetanologyProducts\Demo\Demo2ee\design\..\build\build.sh...
Processing file: E:\MetanologyProducts\Demo\Demo2ee\design\..\build\deploy.sh...
Processing file: E:\MetanologyProducts\Demo\Demo2ee\design\..\build\Defs.mak...
46
44
Document
Output
Completed
Ln 26, Col 0
41

FIG. 3

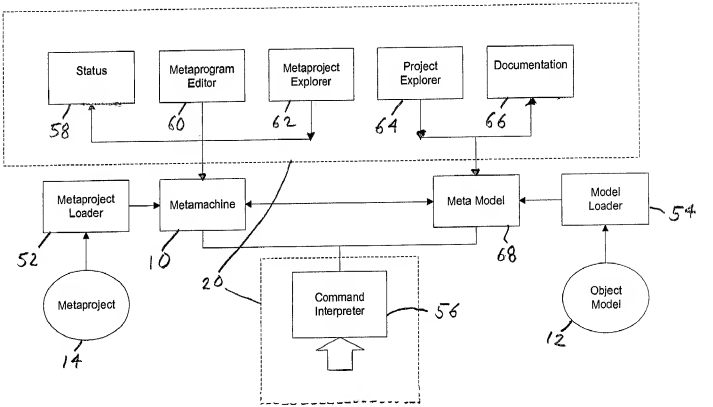
Status
Metaprogram Editor
Metaproject Explorer
Project Explorer
Documentation
58
60
62
64
66
Metaproject Loader
Metamachine
Meta Model
Model Loader
54
52
10
68
20
Command Interpreter
56
Metaproject
14
Object Model
12

FIG. 4

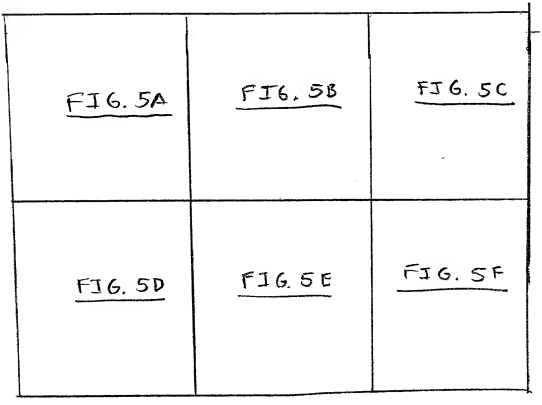

FIG. 5

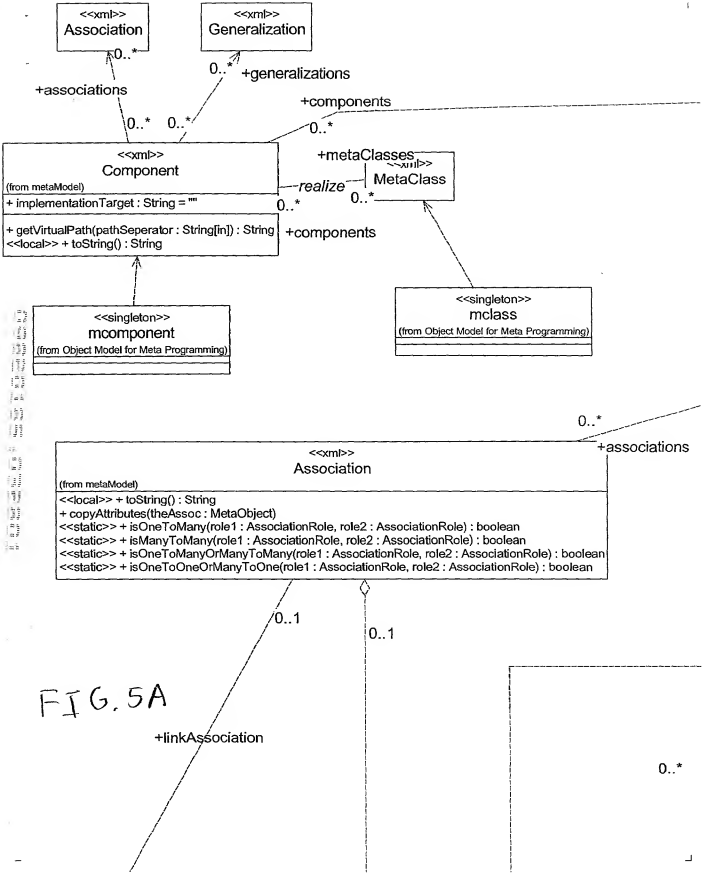
<<xml>>
Association
<<xml>>
Generalization
0..*
0..*
+generalizations
+associations
0..*
0..*
+components
0..*
<<xml>>
Component
(from metaModel)
+ implementationTarget : String = ""
+ getVirtualPath(pathSeperator : String[in]) : String
<<local>> + toString() : String
+metaClasses
<<xml>>
MetaClass
realize
0..*
0..*
+components
<<singleton>>
mcomponent
(from Object Model for Meta Programming)
<<singleton>>
mclass
(from Object Model for Meta Programming)
0..*
+associations
<<xml>>
Association
(from metaModel)
<<local>> + toString() : String
+ copyAttributes(theAssoc : MetaObject)
<<static>> + isOneToMany(role1 : AssociationRole, role2 : AssociationRole) : boolean
<<static>> + isManyToMany(role1 : AssociationRole, role2 : AssociationRole) : boolean
<<static>> + isOneToManyOrManyToMany(role1 : AssociationRole, role2 : AssociationRole) : boolean
<<static>> + isOneToOneOrManyToOne(role1 : AssociationRole, role2 : AssociationRole) : boolean
0..1
0..1
+linkAssociation
0..*

FIG. 5A

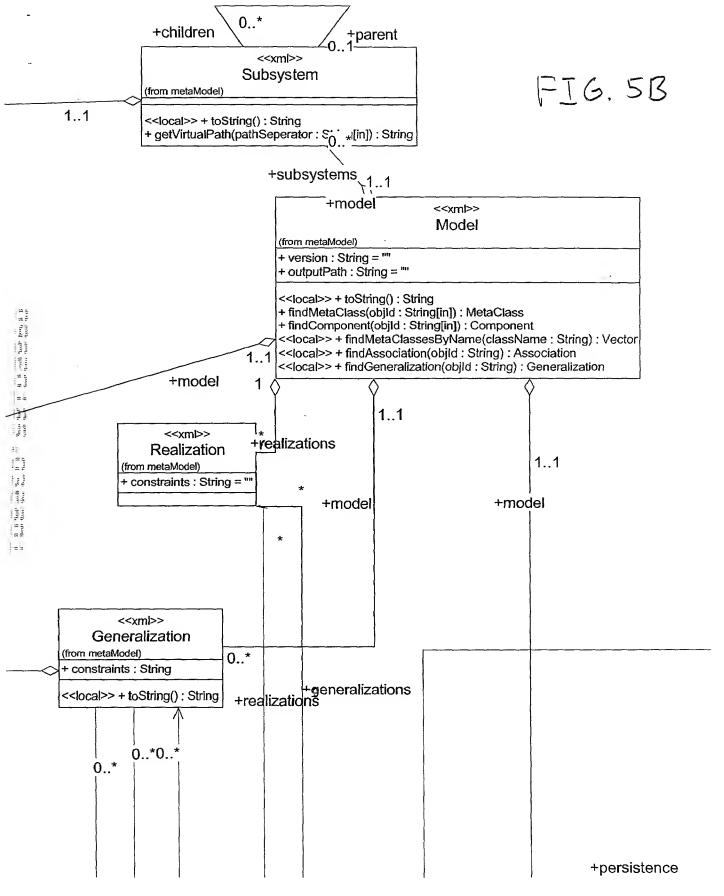
+children
0..*
+parent
0..1
<<xml>>
Subsystem
(from metaModel)
1..1
<<local>> + toString() : String
+ getVirtualPath(pathSeperator : S...[in]) : String
0..*
+subsystems
1..1
+model
<<xml>>
Model
(from metaModel)
+ version : String = ""
+ outputPath : String = ""
<<local>> + toString() : String
+ findMetaClass(objId : String[in]) : MetaClass
+ findComponent(objId : String[in]) : Component
<<local>> + findMetaClassesByName(className : String) : Vector
<<local>> + findAssociation(objId : String) : Association
<<local>> + findGeneralization(objId : String) : Generalization
1..1
+model
1
1..1
<<xml>>
Realization
(from metaModel)
+ constraints : String = ""
*
+realizations
*
+model
1..1
+model
*
<<xml>>
Generalization
(from metaModel)
+ constraints : String
<<local>> + toString() : String
0..*
+realizations
+generalizations
0..*
0..*0..*
+persistence

FIG. 5B

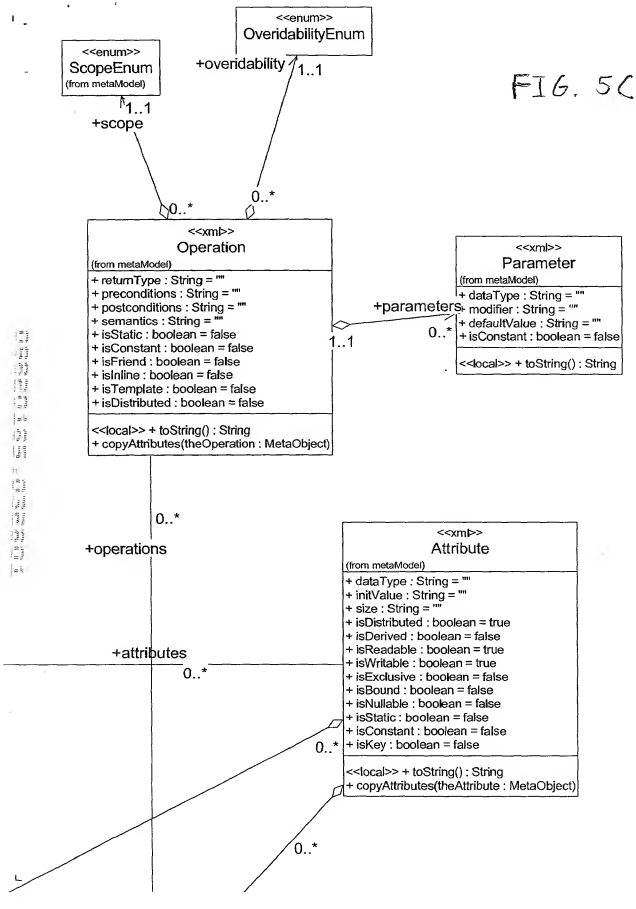
<<enum>>
ScopeEnum
(from metaModel)
<<enum>>
OveridabilityEnum
+overidability
1..1
1..1
+scope
0..*
0..*
<<xml>>
Operation
(from metaModel)
+ returnType : String = ""
+ preconditions : String = ""
+ postconditions : String = ""
+ semantics : String = ""
+ isStatic : boolean = false
+ isConstant : boolean = false
+ isFriend : boolean = false
+ isInline : boolean = false
+ isTemplate : boolean = false
+ isDistributed : boolean = false
<<local>> + toString() : String
+ copyAttributes(theOperation : MetaObject)
+parameters
1..1
0..*
<<xml>>
Parameter
(from metaModel)
+ dataType : String = ""
+ modifier : String = ""
+ defaultValue : String = ""
+ isConstant : boolean = false
<<local>> + toString() : String
0..*
+operations
<<xml>>
Attribute
(from metaModel)
+ dataType : String = ""
+ initValue : String = ""
+ size : String = ""
+ isDistributed : boolean = true
+ isDerived : boolean = false
+ isReadable : boolean = true
+ isWritable : boolean = true
+ isExclusive : boolean = false
+ isBound : boolean = false
+ isNullable : boolean = false
+ isStatic : boolean = false
+ isConstant : boolean = false
+ isKey : boolean = false
<<local>> + toString() : String
+ copyAttributes(theAttribute : MetaObject)
+attributes
0..*
0..*
0..*

FIG. 5C

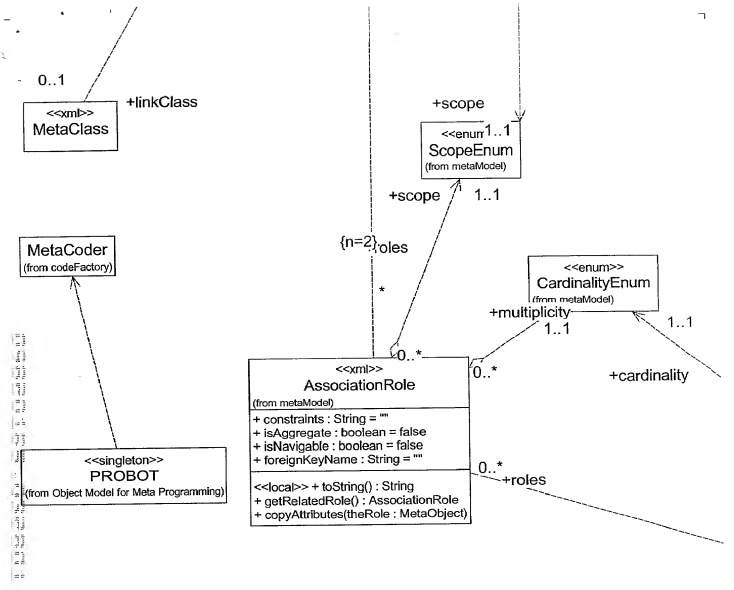
0..1
<<xml>>
MetaClass
+linkClass
+scope
<<enum
1..1
ScopeEnum
(from metaModel)
+scope
1..1
{n=2}
roles
*
MetaCoder
(from codeFactory)
<<enum>>
CardinalityEnum
(from metaModel)
+multiplicity
1..1
1..1
+cardinality
0..*
0..*
<<xml>>
AssociationRole
(from metaModel)
+ constraints : String = ""
+ isAggregate : boolean = false
+ isNavigable : boolean = false
+ foreignKeyName : String = ""
<<local>> + toString() : String
+ getRelatedRole() : AssociationRole
+ copyAttributes(theRole : MetaObject)
0..*
+roles
<<singleton>>
PROBOT
(from Object Model for Meta Programming)

FIG. 5D

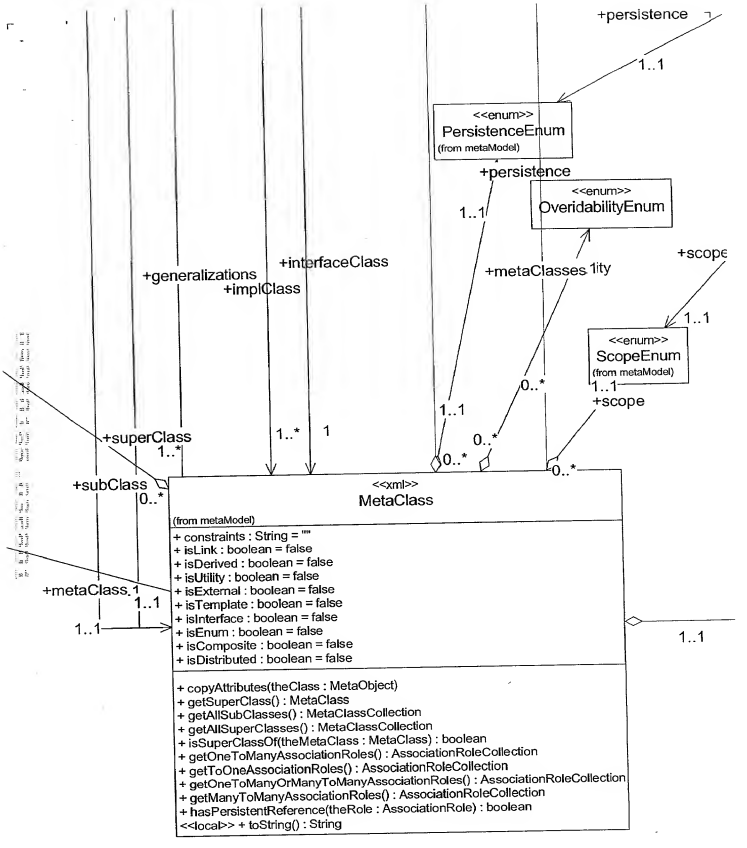
+persistence
1..1
<<enum>>
PersistenceEnum
(from metaModel)
+persistence
1..1
<<enum>>
OveridabilityEnum
+scope
1..1
<<enum>>
ScopeEnum
(from metaModel)
1..1
+scope
+generalizations
+implClass
+interfaceClass
+metaClasses
+superClass
1..*
1..*
1
1..1
0..*
0..*
0..*
0..*
+subClass
0..*
<<xml>>
MetaClass
(from metaModel)
+ constraints : String = ""
+ isLink : boolean = false
+ isDerived : boolean = false
+ isUtility : boolean = false
+ isExternal : boolean = false
+ isTemplate : boolean = false
+ isInterface : boolean = false
+ isEnum : boolean = false
+ isComposite : boolean = false
+ isDistributed : boolean = false
+ copyAttributes(theClass : MetaObject)
+ getSuperClass() : MetaClass
+ getAllSubClasses() : MetaClassCollection
+ getAllSuperClasses() : MetaClassCollection
+ isSuperClassOf(theMetaClass : MetaClass) : boolean
+ getOneToManyAssociationRoles() : AssociationRoleCollection
+ getToOneAssociationRoles() : AssociationRoleCollection
+ getOneToManyOrManyToManyAssociationRoles() : AssociationRoleCollection
+ getManyToManyAssociationRoles() : AssociationRoleCollection
+ hasPersistentReference(theRole : AssociationRole) : boolean
<<local>> + toString() : String
+metaClass 1
1..1
1..1
1..1

FIG. 5E

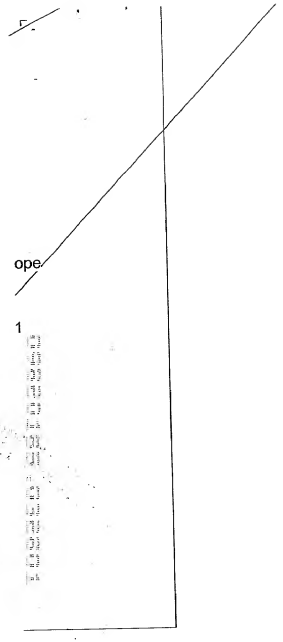

FIG. 5F

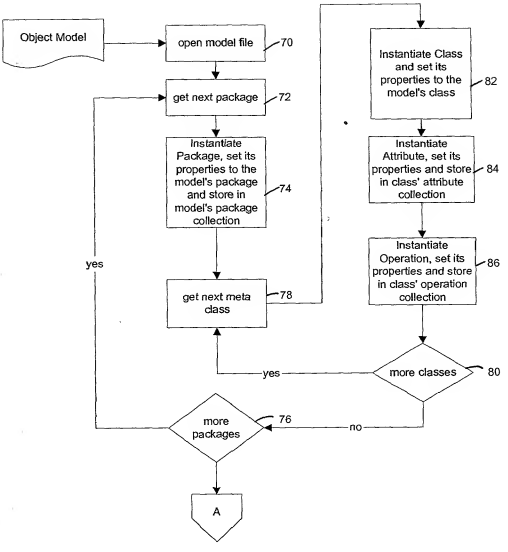
Object Model
open model file
70
get next package
72
Instantiate Package, set its properties to the model's package and store in model's package collection
74
get next meta class
78
Instantiate Class and set its properties to the model's class
82
Instantiate Attribute, set its properties and store in class' attribute collection
84
Instantiate Operation, set its properties and store in class' operation collection
86
more classes
80
more packages
76
yes
yes
no
A

Fig. 6A

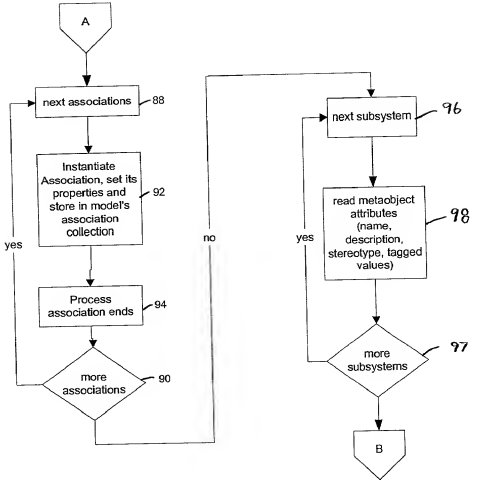
A
next associations
88
Instantiate Association, set its properties and store in model's association collection
92
Process association ends
94
more associations
90
yes
no
next subsystem
96
read metaobject attributes (name, description, stereotype, tagged values)
98
more subsystems
97
yes
B

Fig. 6B

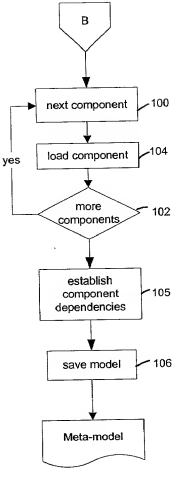
B
next component
100
load component
104
yes
more components
102
establish component dependencies
105
save model
106
Meta-model

Fig. 6C

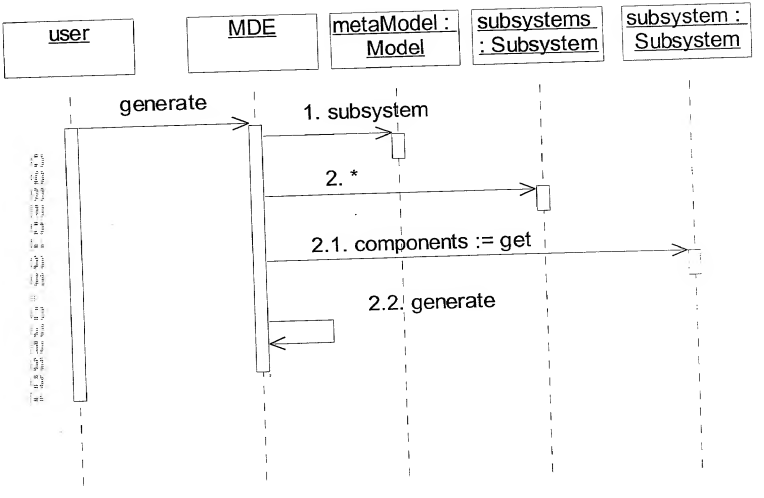
user
MDE
metaModel : Model
subsystems : Subsystem
subsystem : Subsystem
generate
1. subsystem
2. *
2.1. components := get
2.2. generate

Fig. 7

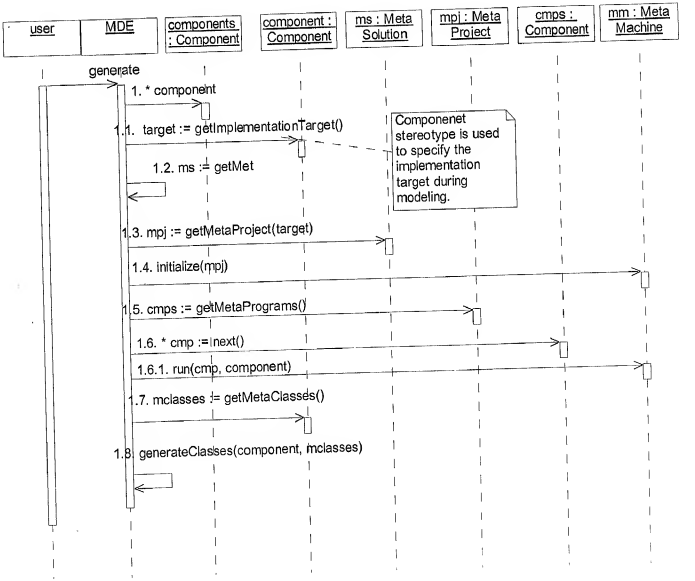
user
MDE
components : Component
component : Component
ms : Meta Solution
mpj : Meta Project
cmps : Component
mm : Meta Machine
generate
1. * component
1.1. target := getImplementationTarget()
Componenet stereotype is used to specify the implementation target during modeling.
1.2. ms := getMet
1.3. mpj := getMetaProject(target)
1.4. initialize(mpj)
1.5. cmps := getMetaPrograms()
1.6. * cmp := next()
1.6.1. run(cmp, component)
1.7. mclasses := getMetaClasses()
1.8. generateClasses(component, mclasses)

Fig. 8

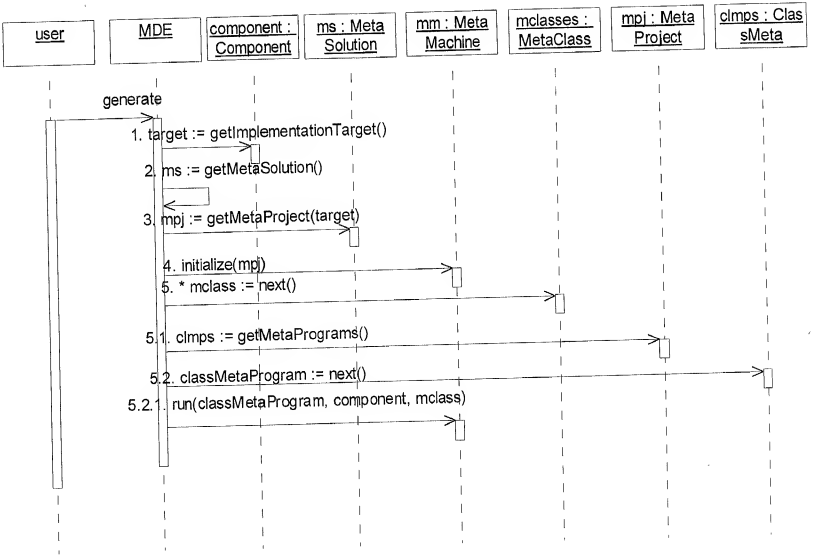
user
MDE
component : Component
ms : Meta Solution
mm : Meta Machine
mclasses : MetaClass
mpj : Meta Project
clmps : ClassMeta
generate
1. target := getImplementationTarget()
2. ms := getMetaSolution()
3. mpj := getMetaProject(target)
4. initialize(mpj)
5. * mclass := next()
5.1. clmps := getMetaPrograms()
5.2. classMetaProgram := next()
5.2.1. run(classMetaProgram, component, mclass)

Fig. 9

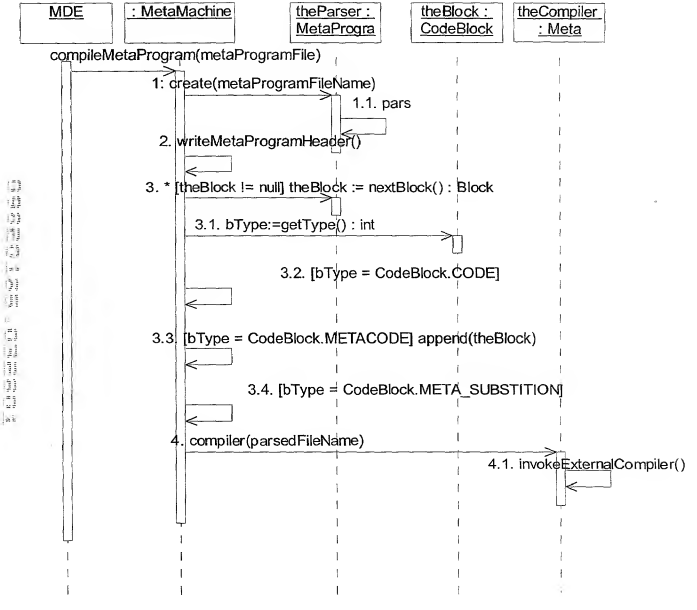
MDE
: MetaMachine
theParser : MetaProgra
theBlock : CodeBlock
theCompiler : Meta
compileMetaProgram(metaProgramFile)
1: create(metaProgramFileName)
1.1. pars
2. writeMetaProgramHeader()
3. * [theBlock != null] theBlock := nextBlock() : Block
3.1. bType:=getType() : int
3.2. [bType = CodeBlock.CODE]
3.3. [bType = CodeBlock.METACODE] append(theBlock)
3.4. [bType = CodeBlock.META_SUBSTITION]
4. compiler(parsedFileName)
4.1. invokeExternalCompiler()

Fig. 10

System Enumerations

PersistenceEnum

Contains enumerations for the persistency of model elements in an object modeling language.

Name	Value	Description
PERSISTENT	15	Indicates a persistent model element.
TRANSIENT	16	Indicates a transient model element.

ScopeEnum

Contains enumerations for the scope a model element may have in an object modeling language.

Name	Value	Description
PUBLIC	11	Indicates public scope.
PROTECTED	12	Indicates protected scope.
PRIVATE	13	Indicates private scope.
IMPLEMENTATION	14	Indicates implementation scope.

CardinalityEnum

Contains enumerations for the multiplicity of an association in an object modeling language.

Name	Value	Description
EXACTLY_ONE	1	Indicates a multiplicty of one (required).
MANY	2	Indicates multiplity of many (e.g., * or m..n).
OPTIONAL	3	Indicates optional multiplicity (e.g., 0, 0..n).
ONE_OR_MORE	4	Indicates multiplicity of at least one (e.g., 1, 1..n).
ZERO_OR_MORE	5	Indicates optional multiplicy (e.g., 0..n,).

OveridabilityEnum

Contains enumerations for the conditions of a model element that allow the model element to be overridden.

Name	Value	Description
VIRTUAL	6	Indicates a virtual model element.
LEAF	7	Indicates a class is a leaf in an inhertance hierarchy.
ABSTRACT	8	Indicates the model element is abstract.
CONCRETE	9	Indicates the class is concrete.
EXTENSIBLE	10	Indicates a model element can be extended (inherited from).

ContainmentEnum

Fig. 11 A

Contains enumerations for the different containment mechanisms between model elements in an object modeling language.

Name	Value	Description
VALUE	20	Indicates containment by value.
REFERENCE	21	Indicates containment by reference.
UNKNOWN_CONTAINMENT	22	Indicates unknown containment method.

Fig. 11 B

Class Attribute

Object
|
+---MetaObject
|
+---Attribute

public class Attribute
extends MetaObject

Description
An Attribute instance represents an attribute of a class in the object model of a software system.

Properties

Name	Type	Description
dataType	String	The data type of the attribute in the object model of a software system represented by this Attribute.
initValue	String	The initial value of the attribute in the object model of the software system represented by this Attribute.
size	String	The size of the attribute in the object model of a software system represented by this Attribute.
isDistributed	boolean	Depricated. True if the attribute in the object model of a software system represented by this Attribute is distributed. The user defined extensions of the object modeling language are used to provide the indication.
isDerived	boolean	True if the attribute in the object model of a software system represented by this Attribute is derived.
isReadable	boolean	True if the attribute in the object model of a software system represented by this Attribute is readable. The user defined extensions of the object modeling language provide the indication.
isWritable	boolean	True if the attribute in the object model of a software system represented by this Attribute is writeable. The user defined extensions of the object modeling language provide the indication.
isExclusive	boolean	Depricated. True if the attribute in the object model of a software system represented by this Attribute is exclusive. The user defined extensions of the object modeling language provide the indication.
isBound	boolean	Depricated. True if the attribute in the object model of a software system represented by this

Fig. 12 A

		Attribute is bound. The user defined extensions of the object modeling language provide the indication.
isNullable	boolean	True if the attribute in the object model of a software system represented by this Attribute is nullable. The user defined extensions of the object modeling language provide the indication.
isStatic	boolean	True if the attribute in the object model of a software system represented by this Attribute is static. Static attributes have class scope. Often, the user defined extensions of the object modeling language provide the indication.
isConstant	boolean	True if the attribute in the object model of a software system represented by this Attribute is constant. The user defined extensions of the object modeling language provide the indication.
isKey	boolean	True if the attribute in the object model of a software system represented by this Attribute is a key. A key is used to uniquely identify objects of the class in the object model of the software system represented by the MetaClass instance that contains this Attribute. The user defined extensions of the object modeling language provide the indication.
scope	int - values defined in class ScopeEnum	Indicates the scope of the attribute in the object model of a software system represented by this Attribute.
persistence	int - values defined in class PersistenceEnum	Indicates the persistence of the attribute in the object model of a software system represented by this Attribute.

Constructors

Name	Description
Attribute()	Constructs an empty Attribute object.

Accessors / Mutators

Name	Description
getDataType()	Return the value of the dataType property.
setDataType(String)	Set the value of the dataType property.
getInitValue()	Return the value of the initValue property.
setInitValue(String)	Set the value of the initValue property.
getSize()	Return the value of the size property.
setSize(String)	Set the value of the size property.
getIsDistributed()	Return the value of the isDistributed property.
setIsDistributed(String)	Set the value of the isDistributed property.
getIsDerived()	Return the value of the isDerived property.
setIsDerived(String)	Set the value of the isDerived property.

Fig. 12 B

getIsReadable()	Return the value of the isReadable property.
setIsReadable(String)	Set the value of the isReadable property.
getIsWritable()	Return the value of the isWritable property.
setIsWritable(String)	Set the value of the isWritable property.
getIsExclusive()	Return the value of the isExclusive property.
setIsExclusive(String)	Set the value of the isExclusive property.
getIsBound()	Return the value of the isBound property.
setIsBound(String)	Set the value of the isBound property.
getIsNullable()	Return the value of the isNullable property.
setIsNullable(String)	Set the value of the isNullable property.
getIsStatic()	Return the value of the isStatic property.
setIsStatic(String)	Set the value of the isStatic property.
getIsConstant()	Return the value of the isConstant property.
setIsConstant(String)	Set the value of the isConstant property.
getIsKey()	Return the value of the isKey property.
setIsKey(String)	Set the value of the isKey property.
getScope()	Return the value of the scope property.
setScope(String)	Set the value of the scope property.
getPersistence()	Return the value of the persistence property.
setPersistence(String)	Set the value of the persistence property.

Associated Objects

Name	Description
getMetaClass()	Returns an object of type MetaClass.

Methods

Name	Description
toString()	Returns a String representation of this Attribute.
copyAttributes()	Copies the attributes of the given Attribute to this Attribute.

Fig. 12 C

Class AssociationRole

Object
|
+---MetaObject
|
+---AssociationRole

public class AssociationRole
extends MetaObject

Description
An AssociationRole instance represents an association end in the object model of a software system.

Properties

Name	Type	Description
constraints	String	The constraints of the association end in the object model of the software system represented by this AssociationRole.
isAggregate	boolean	Indicates the association end in the object model of the software system represented by this AssociationRole is an aggregate.
isNavigable	boolean	Indicates the association end in the object model of the software system represented by this AssociationRole is navigable.
foreignKeyName	String	The name of the foreign key used by objects of the class in the object model of a software system represented by the MetaClass instance of this AssociationRole to access objects on the other side of the association end.
multiplicity	int - values defined in class CardinalityEnum	The multiplicity of the association end in the object model of a software system represented by this AssociationRole.
scope	int - values defined in class ScopeEnum	The scope of the association end in the object model of a software system represented by this AssociationRole.
containment	int - values defined in class ContainmentEnum	The containment of the association end in the object model of a software system represented by this AssociationRole.

Constructors

Name	Description
AssociationRole()	Constructs an empty AssociationRole object.

Accessors / Mutators

Fig. 13 A

Name	Description
getConstraints()	Return the value of the constraints property.
setConstraints(String)	Set the value of the constraints property.
getIsAggregate()	Return the value of the isAggregate property.
setIsAggregate(String)	Set the value of the isAggregate property.
getIsNavigable()	Return the value of the isNavigable property.
setIsNavigable(String)	Set the value of the isNavigable property.
getForeignKeyName()	Return the value of the foreignKeyName property.
setForeignKeyName(String)	Set the value of the foreignKeyName property.
getMultiplicity()	Return the value of the multiplicity property.
setMultiplicity(String)	Set the value of the multiplicity property.
getScope()	Return the value of the scope property.
setScope(String)	Set the value of the scope property.
getContainment()	Return the value of the containment property.
setContainment(String)	Set the value of the containment property.

Associated Objects

Name	Description
getMetaClass()	Returns an object of type MetaClass. The MetaClass instance representing the class in the object model of a software system that participates in the association end in the model represented by this AssociationRole.
getAssociation()	Returns an object of type Association.

Methods

Name	Description
toString()	Returns a String representation of this AssociationRole.
getRelatedRole()	Returns the instance of AssociationRole that represents the other association end in the same association as the association end in the object model of a software system represented by this AssociationRole.
copyAttributes()	Copies the attributes values of the given AssociationRole to this AssociationRole.

Fig. 13 B

Class Component

Object
|
+---MetaObject
|
+---Component

public class Component
extends MetaObject

Description
A Component instance represents a component in the object model of a software system.

Properties

Name	Type	Description
implementationTarget	String	The implementation target of the component in the object model of a software system represented by this Component. The user-defined extensions of the object modeling language are used to define the implementation target. The metamachine binds the Component to the metaproject whose name matches the Components implementation target.

Constructors

Name	Description
Component()	Constructs an empty Component object.

Accessors / Mutators

Name	Description
getImplementationTarget()	Return the value of the implementationTarget property.
setImplementationTarget(String)	Set the value of the implementationTarget property.

Associated Objects

Name	Description
getMetaClasses()	Returns an object of type MetaClass. The MetaClass instances that correspond to the classes in the object model of a software system that belong to the component in the model that is represented by this Component.
getSubsystem()	Returns an object of type Subsystem.
getAssociations()	Returns an object of type Association. The Association instances that represent the associations in the object model of a software system that belong to the component in the model that is represented by this Component.
getGeneralizations()	Returns an object of type Generalization.

Fig. 14 A

	The Generalization instances that represent the generalizations in the object model of a software system that belong to the component in the model that is represented by this Component.
getComponents()	Returns an object of type Component.
getDependencies()	Returns an object of type Component. The Component instances that represent the components in the object model of a software system on which the component in the model that is represented by this Component depends.

Methods

Name	Description
getVirtualPath()	Returns the virtual path to the component, with the nodes in the path separated by the given separator.
toString()	

Fig. 14 B

Class Generalization

Object
|
+---MetaObject
|
+---Generalization

public class Generalization
extends MetaObject

Description
An Generalization instance represents a generalization in the object model of a software system. A generalization exists whenever one class inherits from another class.

Properties

Name	Type	Description
constraints	String	Contains the constraints of the generalization in the object model of a software system represented by this Generalization.
scope	int - values defined in class ScopeEnum	Indicates the scope of the generalization in the object model of a software system represented by this Generalization.

Constructors

Name	Description
Generalization()	Constructs an empty Generalization object.

Accessors / Mutators

Name	Description
getConstraints()	Return the value of the constraints property.
setConstraints(String)	Set the value of the constraints property.
getScope()	Return the value of the scope property.
setScope(String)	Set the value of the scope property.

Associated Objects

Name	Description
getMetaClasss()	Returns an object of type MetaClass.
getSuperClass()	Returns an object of type MetaClass. The MetaClass instance corresponding to the class in the object model of a software system that is the super class of the generalization in the model represented by this Generalization.
getSubClass()	Returns an object of type MetaClass. The MetaClass instance corresponding to the class in the object model of a software system that is the subclass of the generalization in the model represented by this Generalization.

Fig. 15 A

getModel()	Returns an object of type Model. Represents the object model of a software system to which the generalization in the model represented by this Generalization belongs.
getComponents()	Returns an object of type Component.

Methods

Name	Description
toString()	Returns a String representation this Generalization.

Fig. 15 B

Class MetaClass

Object
|
+---MetaObject
|
+---MetaClass

public class MetaClass
extends MetaObject

Description
A MetaClass instance represents a class in an object model of a software system.

Properties

Name	Type	Description
constraints	String	Constraints store the constraints of the class in the object model of the software system.
isLink	boolean	True if the class in the object model of the software system is an association class.
isDerived	boolean	
isUtility	boolean	
isExternal	boolean	True if the user-defined extensions of the object modeling language are used to indicate that the class in the object model of the software system is external. In general, external classes are used by the modeled software system, but they are provided by a different software system.
isTemplate	boolean	True if the user-defined extensions of the object modeling language are used to indicate that the class in the object model of the softwaer system is a template.
isInterface	boolean	True if the user-defined extensions of the object modeling language are used to indicate that the class in the object model of the software system is an interface. Interfaces define the attributes and operations that must be implemented by a class that realizes the interface.
isEnum	boolean	True if the user-defined extensions of the object modeling language are used to indicate that the class in the object model of the software system is an enumeration. Enumerations are classes that have only initialized class-scope attributes whose values never change (readonly). Generally, they are used to restrict the value of other instance-scope attributes in the software system.
isComposite	boolean	

Fig. 16 A

isDistributed	boolean	
overidability	int - values defined in class OveridabilityEnum	Indicates the overridability of the class in the object model of a software system represented by this MetaClass.
persistence	int - values defined in class PersistenceEnum	Indicates the persistence of the class in the object model of a software system represented by this MetaClass.
scope	int - values defined in class ScopeEnum	Indicates the scope of the class in the object model of a software system represented by this MetaClass.
cardinality	int - values defined in class CardinalityEnum	Indicates the type of the class in the object model of a software system represented by this MetaClass. A regular class allowing multiple instances; a singleton allowing one instance; or, a one with all class-scope attributes and methods (e.g., static).

Constructors

Name	Description
MetaClass()	Constructs an empty MetaClass object.

Accessors / Mutators

Name	Description
getConstraints()	Return the value of the constraints property.
setConstraints(String)	Set the value of the constraints property.
getIsLink()	Return the value of the isLink property.
setIsLink(String)	Set the value of the isLink property.
getIsDerived()	Return the value of the isDerived property.
setIsDerived(String)	Set the value of the isDerived property.
getIsUtility()	Return the value of the isUtility property.
setIsUtility(String)	Set the value of the isUtility property.
getIsExternal()	Return the value of the isExternal property.
setIsExternal(String)	Set the value of the isExternal property.
getIsTemplate()	Return the value of the isTemplate property.
setIsTemplate(String)	Set the value of the isTemplate property.
getIsInterface()	Return the value of the isInterface property.
setIsInterface(String)	Set the value of the isInterface property.
getIsEnum()	Return the value of the isEnum property.
setIsEnum(String)	Set the value of the isEnum property.
getIsComposite()	Return the value of the isComposite property.
setIsComposite(String)	Set the value of the isComposite property.
getIsDistributed()	Return the value of the isDistributed property.
setIsDistributed(String)	Set the value of the isDistributed property.
getOveridability()	Return the value of the overidability property.
setOveridability(String)	Set the value of the overidability property.

Fig. 16 B

getPersistence()	Return the value of the persistence property.
setPersistence(String)	Set the value of the persistence property.
getScope()	Return the value of the scope property.
setScope(String)	Set the value of the scope property.
getCardinality()	Return the value of the cardinality property.
setCardinality(String)	Set the value of the cardinality property.

Associated Objects

Name	Description
getHoldingPackage()	Returns an object of type Package. The holdingPackage represents the package in the object model of a software system to which the class represented by this MetaClass belongs.
getAttributes()	Returns an object of type Attribute. The attributes of the class in the object model of a software system represented by this MetaClass.
getOperations()	Returns an object of type Operation. The operations of the class in the object model of the software system represented by this MetaClass.
getGeneralizations()	Returns an object of type Generalization. The generalizations the class in the object model of a software system represented by this MetaClass participates in as a subclass.
getGeneralizations()	Returns an object of type Generalization.
getGeneralizations()	Returns an object of type Generalization.
getRoles()	Returns an object of type AssociationRole. The association ends of the class in the object model of a software system represented by this MetaClass participates.
getComponents()	Returns an object of type Component. The components in which the class in the object model of a software system represented by this MetaClass is realized.
getLinkAssociation()	Returns an object of type Association. The association of which the class in the object model of a software system represented by this MetaClass is an association class.
getRealizations()	Returns a collection of Realization objects. The realizations in which the class in the object model of a software system represented by this MetaClass participates as implementing the corresponding interface.
getRealizations()	Returns a collection of Realization objects.
getRealizations()	Returns a collection of Realization objects.
getModel()	Returns an object of type Model. Represents the object model of a software system to which the class in the model represented by this MetaClass belongs.

Methods

Name	Description

Fig. 16 C

copyAttributes()	Copies the attributes from the given class to this class.
getSuperClass()	If the class in the model of the software system inherits from another class in the model, return the metaclass instance that represents the superclass. Otherwise, return null.
getAllSubClasses()	Returns all the metaclass instances that represent classes in the model of the software system that inherit from the class in the model represented by this MetaClass.
getAllSuperClasses()	Returns all the metaclass instances that represent classes in the model of the software system from which the class in the model represented by this MetaClass inherits .
isSuperClassOf()	Return true if the class in the software system's model represented by this MetaClass instance inherits from the class in the model represented by the given MetaClass instance. Otherwise, return false.
getOneToManyAssociationRoles()	Returns the meta roles representing the one-to-one and one-to-many roles in the model of the software system in which the class in the model represented by this instance of the MetaClass participates.
getToOneAssociationRoles()	Returns the meta roles representing the one-to-one and many-to-one roles in the model of the software system in which the class in the model represented by this instance of the MetaClass participates.
getOneToManyOrManyToManyAssociationRoles()	Returns the meta roles representing the one-to-many and many-to-many roles in the model of the software system in which the class in the model represented by this instance of the MetaClass participates.
getManyToManyAssociationRoles()	Returns the meta roles representing the many-to-many roles in the model of the software system in which the class in the

Fig. 16 D

	model represented by this instance of the MetaClass participates.
hasPersistentReference()	Returns true if the the class in the model of the software system represented by this MetaClass contains a persistent reference to the class in the model represented by the MetaClass contained in the given Roler. Otherwise, return false.
toString()	Returns a String representation of the MetaClass.

Fig. 16 E

Class MetaObject

Object
|
+---MetaObject

public class MetaObject

Description
The base class of most objects in the meta model.

Properties

Name	Type	Description
objId	String	The object id of the model element. The object id uniquely identifies a model element in an object model of a software system.
name	String	The name of the model element.
description	String	A description of the model element. The user-defined extensions of an object modeling language are often used to set description's value.
stereotype	String	The stereotype of the model element. Stereotypes are user-defined extensions of UML.

Constructors

Name	Description
MetaObject()	Constructs an empty MetaObject object.

Accessors / Mutators

Name	Description
getObjId()	Return the value of the objId property.
setObjId(String)	Set the value of the objId property.
getName()	Return the value of the name property.
setName(String)	Set the value of the name property.
getDescription()	Return the value of the description property.
setDescription(String)	Set the value of the description property.
getStereotype()	Return the value of the stereotype property.
setStereotype(String)	Set the value of the stereotype property.

Associated Objects

Name	Description
getTags()	Returns a collection of Tag objects. The tags of the model element in the object model of a software system represented by this MetaObject.

Methods

Fig. 17 A

Name	Description
getTaggedValue()	Returns the value of the given tag.
setTaggedValue()	Sets the given tag to the given value.
getTags()	Return the tagged values of this MetaObject.
copyAttributes()	Copies the attributes values of the given MetaObject to this MetaObject.
hasStereotype()	Returns true if the MetaObject has the given stereotype.
toString()	Returns a String representation of this MetaObject.

Fig. 17 B

Class Model

Object
|
+---MetaObject
|
+---Model

public class Model
extends MetaObject

Description
The root element of an object model of a software system.

Properties

Name	Type	Description
version	String	The version of the model.
outputPath	String	The pathname of where metaprograms generate output.

Constructors

Name	Description
Model()	Constructs an empty Model object.

Accessors / Mutators

Name	Description
getVersion()	Return the value of the version property.
setVersion(String)	Set the value of the version property.
getOutputPath()	Return the value of the outputPath property.
setOutputPath(String)	Set the value of the outputPath property.

Associated Objects

Name	Description
getMetaClasses()	Returns an object of type MetaClass. The MetaClass instances representing the classes in the object model of a software system represented by this Model.
getSubsystems()	Returns an object of type Subsystem. The Subsystem instances representing the subsystems in the object model of a software system represented by this Model.
getPackages()	Returns an object of type Package. The Package instances representing the packages in the object model of a software system represented by this Model.
getAssociations()	Returns an object of type Association. The Association instances representing the associations in the object model of a software system represented by this Model.
getGeneralizations()	Returns an object of type Generalization. The Generalization instances representing the generalizations in

Fig. 18 A

	the object model of a software system represented by this Model.
getRealizations()	Returns a collection of Realization objects. The Realization instances representing the realizations in the object model of a software system represented by this Model.
getMetaCoders()	Returns an object of type MetaCoder.

Methods

Name	Description
toString()	Returns a String representation of this model.
findMetaClass()	Returns the instance of MetaClass whose object id matches the given object id. Returns null if no such MetaClass exists.
findComponent()	Returns the Component instance whose object id matches the given object id. Returns null if no such Component instance exists.
findMetaClassesByName()	Returns a collection of MetaClass instances whose names match the given name. Returns null if no such MetaClass instances exist.
findAssociation()	Returns the instance of Association whose object id matches the given object id. Returns null if no such instance exists.
findGeneralization()	Returns the instance of Generalization whose object id matches the given object id. Returns null if no such instance exists.

Fig. 18 B

Class Operation

Object
|
+---MetaObject
|
+---Operation

public class Operation
extends MetaObject

Description
An Operation instance represents a method of a class in the object model of a software system.

Properties

Name	Type	Description
returnType	String	The return type of the method in the object model of the software system represented by this Operation.
preconditions	String	The preconditions of the method of the class in the object model of the software system represented by this Operation.
postconditions	String	The postconditions of the method of the class in the object model of the software system represented by this Operation.
semantics	String	The semantics of the method of the class in the object model of the software system represented by this Operation.
isStatic	boolean	True if the method of the class in the object model of the software system represented by this Operator is static. Static methods have class scope. The user defined extension of the object modeling language are sometimes used to give this indication.
isConstant	boolean	Depricated.
isFriend	boolean	Depricated. Indicates the method of the class in the object model of the software system represented by this Operator is a friend. The user defined extensions of the object modeling language are used to give this indication.
isInline	boolean	Depricated. Indicates that the method of the class in the model of the software system represented by this Operator is inline. The user defined extensions of the object modeling language are used to provide this indication.
isTemplate	boolean	Depricated.

Fig. 19 A

isDistributed	boolean	Depricated. True if the method of the class in the object model of a software system represented by this Operator is distributed. The user defined extensions of the object modeling language are used to provide the indication.
overidability	int - values defined in class OveridabilityEnum	Indicates the overridability of the method in the object model of a software system represented by this Operation.
scope	int - values defined in class ScopeEnum	

Constructors

Name	Description
Operation()	Constructs an empty Operation object.

Accessors / Mutators

Name	Description
getReturnType()	Return the value of the returnType property.
setReturnType(String)	Set the value of the returnType property.
getPreconditions()	Return the value of the preconditions property.
setPreconditions(String)	Set the value of the preconditions property.
getPostconditions()	Return the value of the postconditions property.
setPostconditions(String)	Set the value of the postconditions property.
getSemantics()	Return the value of the semantics property.
setSemantics(String)	Set the value of the semantics property.
getIsStatic()	Return the value of the isStatic property.
setIsStatic(String)	Set the value of the isStatic property.
getIsConstant()	Return the value of the isConstant property.
setIsConstant(String)	Set the value of the isConstant property.
getIsFriend()	Return the value of the isFriend property.
setIsFriend(String)	Set the value of the isFriend property.
getIsInline()	Return the value of the isInline property.
setIsInline(String)	Set the value of the isInline property.
getIsTemplate()	Return the value of the isTemplate property.
setIsTemplate(String)	Set the value of the isTemplate property.
getIsDistributed()	Return the value of the isDistributed property.
setIsDistributed(String)	Set the value of the isDistributed property.
getOveridability()	Return the value of the overidability property.
setOveridability(String)	Set the value of the overidability property.
getScope()	Return the value of the scope property.
setScope(String)	Set the value of the scope property.

Associated Objects

Fig. 19 B

Name	Description
getMetaClass()	Returns an object of type MetaClass.
getParameters()	Returns an object of type Parameter. The Parameter instances representing parameters in the object model of a software system that are parameters to the method in the model represented by this Operation.

Methods

Name	Description
toString()	Returns a String representation of this Operator.
copyAttributes()	Copies the attributes values of the given Operator to this Operator.

Fig. 19 C

Class Package

Object
  |
  +---MetaObject
    |
    +---Package

public class Package
extends MetaObject

Description
A Package instance represents a package in an object model of a software system.

Constructors

Name	Description
Package()	Constructs an empty Package object.

Associated Objects

Name	Description
getMetaClasses()	Returns an object of type MetaClass. The MetaClass instances that correspond to the classes in the object model of a software system that belong to the package in the model that is represented by this Package.
getChildren()	Returns an object of type Package. The Package instances of the packages in the object model of a software system that represent the children packages in the model represented by this Package.
getParent()	Returns an object of type Package. The Package instance of the package in the object model of a software system that represents the parent package in the model represented by this Package.
getModel()	Returns an object of type Model. Represents the object model of a software system to which the package in the model represented by this Package belongs.

Methods

Name	Description
toString()	Returns a String representation of the Package.

Fig. 20

Class Parameter

Object
|
+---MetaObject
|
+---Parameter

public class Parameter
extends MetaObject

Description
A meta parameter represents a parameter to a method in the model of a software system.

Properties

Name	Type	Description
dataType	String	The data type of the parameter in the model of the software system represented by this "meta" parameter.
modifier	String	The modifier of the parameter in the model of the software system represented by this "meta" parameter.
defaultValue	String	The default value of the parameter in the model of the software system represented by this "meta" parameter.
isConstant	boolean	True if the parameter in the model of the software system represented by this "meta" parameter is a constant.

Constructors

Name	Description
Parameter()	Constructs an empty Parameter object.

Accessors / Mutators

Name	Description
getDataType()	Return the value of the dataType property.
setDataType(String)	Set the value of the dataType property.
getModifier()	Return the value of the modifier property.
setModifier(String)	Set the value of the modifier property.
getDefaultValue()	Return the value of the defaultValue property.
setDefaultValue(String)	Set the value of the defaultValue property.
getIsConstant()	Return the value of the isConstant property.
setIsConstant(String)	Set the value of the isConstant property.

Associated Objects

Name	Description
getOperation()	Returns an object of type Operation.

Methods

Fig. 21A

Name	Description
toString()	Returns a string representation of this "meta" parameter.

Fig. 21B

Class Realization

Object
|
+---MetaObject
|
+---Realization

public class Realization
extends MetaObject

Description
A Realization instance represents a realization of an interface by a class in the object model of the software system.

Properties

Name	Type	Description
constraints	String	The constraints of this Realization.

Constructors

Name	Description
Realization()	Constructs an empty Realization object.

Accessors / Mutators

Name	Description
getConstraints()	Return the value of the constraints property.
setConstraints(String)	Set the value of the constraints property.

Associated Objects

Name	Description
getMetaClasss()	Returns an object of type MetaClass.
getInterfaceClass()	Returns an object of type MetaClass. The MetaClass instance representing the class in the object model of a software sytem that defines the interface being realized.
getImplClass()	Returns an object of type MetaClass. The MetaClass instance representing the class in the object model of a software sytem that implements the interface being realized.
getModel()	Returns an object of type Model.

Fig. 22

Class Subsystem

Object
|
+---MetaObject
|
+---Subsystem

public class Subsystem
extends MetaObject

Description
An instance of Subsystem represents a subsystem in the model of the software system. A subsystem is used to group components and typically affects the physical layout of implementation of the software system.

Constructors

Name	Description
Subsystem()	Constructs an empty Subsystem object.

Associated Objects

Name	Description
getChildren()	Returns an object of type Subsystem. The Subsystem instances of the subsystems in the object model of a software system that represent the children subsystems in the model represented by this Subsystem.
getParent()	Returns an object of type Subsystem. The Subsystem instance of the subsystem in the object model of a software system that represents the parent subsystem in the model represented by this Subsystem.
getModel()	Returns an object of type Model. Represents the object model of a software system to which the class in the model represented by this MetaClass belongs.
getComponents()	Returns an object of type Component. The components belonging to the subsystem in the object model of a software system represented by this Subsytem.

Methods

Name	Description
toString()	Returns a String representation of the Subsystem.
getVirtualPath()	Returns the virtual path of the subsystem with each node separated by the given separator. In general, the virtual path of a Subsystem is used to determine the pathname to the generated components that comprise the subsystem.

Fig. 23

Class Tag

Object
|
+---Tag

public class Tag

Description
A Tag instance represents a tagged value bound to a model element in an object model of a software system.

Properties

Name	Type	Description
name	String	The name of the tag.
value	String	The tag's value.

Constructors

Name	Description
Tag()	Constructs an empty Tag object.

Accessors / Mutators

Name	Description
getName()	Return the value of the name property.
setName(String)	Set the value of the name property.
getValue()	Return the value of the value property.
setValue(String)	Set the value of the value property.

Associated Objects

Name	Description
getMetaObject()	Returns an object of type MetaObject.

Fig. 24